CLAIMS

What is claimed is:

1. A method of mapping a plurality of virtual registers to a plurality of physical registers comprising:

providing a plurality of virtual registers, wherein each virtual register comprises physical register address bits; and

providing a status indicator for indicating the status of each virtual register.

2. The method, as recited in claim 1, further comprising:

mapping a virtual register from an old physical register to a new physical register, when the virtual register is a destination virtual register of an instruction being decoded; and

placing an address of the old physical register in an instruction retirement list related to the instruction being decoded if and only if the status indicator indicates that the virtual register is not clean.

3. The method, as recited in claim 2, further comprising saving a physical register address held in a virtual register and a status of the virtual register indicated by the status indicator to a stack and setting to clean at least status of the virtual register.

4. The method, as recited in claim 2, further comprising setting the status of a virtual register to not clean when the virtual register is mapped to a new physical register.

5. The method, as recited in claim 4, wherein the status indicator comprises:

a plurality of status bits, wherein each virtual register of the plurality of virtual registers is associated with at least one of the status bits of the plurality of status bits; and

a comparator, which detects the value in each of the status bits of the plurality of status bits, wherein the values in the status bits associated with the saving of a status on a stack is accomplished by saving a value in at least one status bit on the stack.

6. The method, as recited in claim 1, further comprising designating a plurality of virtual registers of the plurality of virtual registers as virtual local registers.

7. The method, as recited in claim 6, further comprising executing a save command, comprising:

saving the mapping of all virtual local registers onto a stack; and

saving a status as indicated by the status indicator for each of the virtual local registers onto the stack.

8. The method, as recited in claim 7, wherein the save command further comprises setting the status of all virtual local registers to "clean".

9. The method, as recited in claim 8, further comprising executing a restore command, comprising:

popping the mapping of all virtual local registers from the stack to the virtual local registers; and

popping the status of all virtual local registers from the stack.

10. The method, as recited in claim 9, further comprising:

binding a first virtual register of the plurality of virtual registers to a second virtual register of the plurality of virtual registers; and

binding the status of the first virtual register to the second virtual register.

11. The method, as recited in claim 10, wherein the binding comprises placing a physical address stored in the second virtual register in the first virtual register and setting the status of the first virtual register to the status of the second virtual register.

12. The method, as recited in claim 11, wherein the binding further comprises:

saving the mapping of the first virtual register onto the stack cache; and

saving the status of the first virtual local register onto the stack cache.

13. The method, as recited in claim 12, wherein the binding occurs during a call instruction, wherein the call instruction has at least one argument, wherein the first virtual register is used for the at least one argument.

14. The method, as recited in claim 11, wherein subsequent the binding when the first virtual register is a destination register, the first virtual register is assigned a physical register address which is different than a physical register address stored in the second virtual register.

15. The method, as recited in claim 1, wherein the status indicator comprises:

a plurality of status bits, wherein each virtual register of the plurality of virtual registers is associated with at least one of the status bits of the plurality of status bits; and

a comparator, which detects the value in each of the status bits of the plurality of status bits.

16. A processing device, comprising:

a plurality of physical registers;

a plurality of virtual registers, wherein each virtual register comprises physical register address bits; and

a status indicator for indicating a status of each virtual register.

17. The processing device, as recited in claim 16, further comprising:

an instruction decoder for decoding an instruction;

an instruction retirement list;

machine readable code for mapping a virtual register from an old physical register to a new physical register, when the virtual register is a destination virtual register of an instruction being decoded; and

machine readable code for placing an address of the old physical register in an instruction retirement list related to the instruction being decoded if and only if the status indicator indicates that the virtual register is not clean.

18. The processing device, as recited in claim 17, further comprising a stack, wherein a physical register address held in a virtual register and at least one status bit associated with the virtual register of the plurality of virtual registers is saved to the stack and the status of the saved virtual register is set to clean.

19. The processing device, as recited in claim 18, wherein the status indicator comprises:

a plurality of status bits, wherein each virtual register of the plurality of virtual registers is associated with at least one of the status bits of the plurality of status bits; and

a comparator, which detects the value in each of the status bits of the plurality of status bits, wherein the values in the status bits associated with the saving of a status on a stack is accomplished by saving a value in at least one status bit on the stack.

20. The processing device, as recited in claim 19, wherein the status of a virtual register is set to not clean when the virtual register is mapped to a new physical register.